

# Grow Real-Time UML Through Innovation

BY PAUL MOOREHEAD

CEO, ARTISAN SOFTWARE TOOLS LTD.

**Q**uite a buzz is swirling among tool vendors on the subject of industry standards and their role in the evolution of real-time software and system modeling tools. Both in printed publications and on Internet newsgroups, heated debates have been sparked by passionate feelings on what should and shouldn't become a standard for the industry.

Real-time software developers need an intimate knowledge of a system's architecture and temporal behavior, well beyond what the developer of an IS application requires. UML evolved largely to support IS development. These additional demands have driven some to suggest that additional modeling views (beyond UML) are helpful to the development process for real-time systems.

In the May issue of *Embedded Systems Development*, Bruce Powel Douglass expounded the virtues of UML, asserting that it is complete and appropriate by itself for real-time development. He also upholds the role of the Object Management Group (OMG) as the clearinghouse for any extensions to UML. While we disagree about the sufficiency of UML for real-time systems, in the strict sense he is right about the role of the OMG.

There are many reasons why the OMG is structured as a committee. It's up to the members of the OMG to avoid over-complicating the standards for which it's responsible, and to hold each and every member in check. So far, UML is reasonably concise, and is a well-designed notation. I agree with Douglass in his earlier article that UML should be protected from bloat. Identifying

additional real-time design issues that need addressing in tools doesn't mean that we should grow UML in every case. UML needn't become the all-encompassing design notation in order to be an effective standard.

Standards are a helpful means of converging competing approaches to a technical area of interest. UML is a fine notation. Why invent another to



**It is quite acceptable for a tool to support notations that are not part of UML.**

do the same thing? However, if no vendor ever marketed an innovative idea that went beyond the existing standards, there would be no competing ideas for the standards committee to consolidate! Furthermore, imagine how slowly technology would have developed if all vendors waited for standards committee action before introducing an innovation—whether it be the evolution of the automobile, the computer, or software design notation. Had there not first been Booch's notation, Rumbaugh's OMT, and Jacobsen's Objectory—none of which was approved by a standards committee—would there be a UML today?

Software-tool vendors continue to innovate by providing solutions for real-time system development years ahead of standards committee conclusions on real-time extensions to UML. We need to continue to support OMG activity and OMG standards as they emerge. However, vendors have customers with real-world development problems that won't wait for standards approval—problems that can be solved today but aren't sufficiently addressed by UML. Can UML be contorted to provide a view of multitasking? Yes, but a system's tasking model can be conveyed in better ways.

When considering changes to the UML standard, people have voiced concerns that the core of UML may become corrupted if certain extensions are permitted. Other extensions offer helpful notations by contributing another view of existing modeling constructs. The extra view might be used during the development process to evolve the primary UML views. One example of this kind of extension is a concurrency model used to show tasks and intertask communication primitives. It guides users to build appropriate classes and relationships in the class model to support the underlying RTOS. Perhaps "extension" is the wrong word, and is causing Mr. Douglass' angst. These are "companion" views of a system, providing ancillary support that are complementary to UML.

It's quite acceptable for a tool to support notations that aren't part of UML. There's no particular need for these extensions to be forced into the standard. The only requirement is for the tool to support the relevant notations that are defined in UML. What shouldn't be considered acceptable is for a tool to support a notation that conflicts with the UML standard and still make claims that it complies with UML.

The point is that we want the UML to remain a clean and general-purpose standard. Making the UML standard include best-practice techniques from the real-time industry is preferable. However, any extension should be considered conservatively, and added only if it doesn't corrupt

the core of the existing standard.

Innovative solutions can consist of standard UML views and additional system and tasking views that aren't yet an OMG supported standard. Foremost is that they be easy to adopt, and more importantly, they must work. We don't believe that, axiomatically, all real-time embedded systems and software modeling should be constrained to fit today's UML, we believe in doing what makes sense.

That brings us to UML's *scope*. The debate in these pages appears to be partly over whether UML ought to be a real-time specific notation or not. Originally intended to model use cases and object architecture, it is well accepted as an appropriate foundation for modeling real-time systems. However, there are more straightforward notations and methods for depicting specific real-time attributes.

The dilemma facing the OMG is whether UML in its current form is sufficient for real-time (as Mr. Douglass insists), to eventually adopt standards for real-time modeling, or to define the scope of UML as system-generic only, leaving the real-time details to vendor-specific implementations. Experience has convinced us that developers using UML-only tools are very frustrated trying to express everyday real-time constructs. We see the marketplace voting with its dollars for solutions better tailored to real-time design problems.

It's fascinating and dismaying to watch religious wars emerge over what is and what is not a "blessed" standard. It recalls images of the late 1980s when CASE vendors and their followers engaged in heated debates over whether rectangular boxes on a diagram should have rounded corners or square corners, as if it really mattered.

Standards should be used wherever possible. But the industry is driven by pragmatists, not academics, so vendors must measure the "correctness" of their offerings by customer success, not ideological purity. Consequently, we shy away from religious arguments over the "correct" shape of the hammer and prefer not to wait for eventual "Vatican approval" of hammers, especially when the customer just wants to drive the nail today. **ESD**

**New!** **1600 MIPS**

**Single-Board Computer** **Modular I/O**

**SBC62** | *A dramatically new design, using modular I/O and the state-of-the-art C6201 DSP for the ultimate in speed and versatility.*

- Full Windows '95/NT development software support
- Stand Alone Processor card 160 mm x 100 mm
- Dual I/O expansion sites for ultra-fast, user configurable I/O
- Analog I/O
- Digital and serial I/O
- Performance audio
- Telephony
- Motion control
- Fast data acquisition

Call: 818.865.6150  
fax: 818.879.1770  
www.innovative-dsp.com

**Innovative Integration**  
... real time. solutions!

**It takes numerous things to get into a great design magazine...**

**PENTON'S**

**EMBEDDED SYSTEMS DEVELOPMENT**

**TOTAL SOLUTIONS FOR EMBEDDED AND REAL-TIME DESIGN**

**... a great design idea is one of them!**